Contents

Managing Gateways with Multiple Revisions

5 minute read page test

This feature is actively in development and is considered experimental.

With a single `IstioOperator` CR, any gateways defined in the CR (including the `istio-ingressgateway` installed in the default profile) are upgraded in place, even when the canary control plane method is used. This is undesirable because gateways are a critical component affecting application uptime. They should be upgraded last, after the new control and data plane versions are verified to be working.

This guide describes the recommended way to upgrade gateways by defining and managing them in a separate `IstioOperator` CR, separate from the one used to install and manage the control plane.

To avoid problems with `.` (dot) not being a valid character in some Kubernetes paths, the revision

name should not include `.` (dots).

Istioctl

This section covers the installation and upgrade of a separate control plane and gateway using `istioctl`. The example demonstrates how to upgrade Istio 1.8.0 to 1.8.1 using canary upgrade, with gateways being managed separately from the control plane.

Installation with `istioctl`

1. Ensure that the main `IstioOperator` CR has a name and does not install a gateway:

```
# filename: control-plane.yaml
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  name: control-plane # REQUIRED
spec:
  profile: minimal
```

2. Create a separate `IstioOperator` CR for the gateway(s), ensuring that it has a name and has the `empty` profile:

```
# filename: gateways.yaml
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  name: gateways # REQUIRED
spec:
  profile: empty # REQUIRED
  components:
    ingressGateways:
      - name: istio-ingressgateway
        enabled: true
```

3. Install the `CR`s:

```
$ istio-1.8.0/bin/istioctl install -n istio-sys
tem -f control-plane.yaml --revision 1-8-0
$ istio-1.8.0/bin/istioctl install -n istio-sys
tem -f gateways.yaml --revision 1-8-0
```

Istioctl install and the operator track resource ownership through labels for both the revision and owning CR name. Only resources whose name and revision labels match the `IstioOperator` CR passed to `istioctl` install/operator will be affected by any changes to the CR - all other resources in the cluster will be ignored. It is important to make sure that each `IstioOperator` installs components that do not overlap with another `IstioOperator` CR, otherwise the two CR's will cause controllers or `istioctl` commands to interfere with each other.

Upgrade with `istioctl`

Let’s assume that the target version is 1.8.1.

1. Download the Istio 1.8.1 release and use the `istioctl` from that release to install the Istio 1.8.1 control plane:

   ```
   $ istio-1.8.1/bin/istioctl install -f control-plane.yaml --revision 1-8-1
   ```

   (Refer to the canary upgrade docs for more details on steps 2-4.)
2. Verify that the control plane is functional.
3. Label workload namespaces with istio.io/rev=1-8-1 and restart the workloads.
4. Verify that the workloads are injected

with the new proxy version and the cluster is functional.

5. At this point, the ingress gateway is still 1.8.0. You should see the following pods running:

```
$ kubectl get pods -n istio-system --show-label
s

NAME                                          READY
 STATUS    RESTARTS   AGE   LABELS
istio-ingressgateway-65f8bdd46c-d49wf   1/1
 Running   0          21m   service.istio.io/ca
nonical-revision=1-8-0 ...
istiod-1-8-0-67f9b9b56-r22t5            1/1
 Running   0          22m   istio.io/rev=1-8-0
...
istiod-1-8-1-75dfd7d494-xhmbb           1/1
 Running   0          21s   istio.io/rev=1-8-1
...
```

As a last step, upgrade any gateways in the cluster to the new version:

```
$ istio-1.8.1/bin/istioctl install -f gateways.
yaml --revision 1-8-1
```

6. Delete the 1.8.0 version of the control plane:

```
$ istio-1.8.1/bin/istioctl x uninstall --revisi
on 1-8-0
```

Operator

This section covers the installation and upgrade of a separate control plane and gateway using the Istio operator. The example demonstrates how to upgrade Istio 1.8.0 to 1.8.1 using canary upgrade, with gateways being managed separately from the control plane.

Installation with

operator

1. Install the Istio operator with a revision into the cluster:

```
$ istio-1.8.0/bin/istioctl operator init --revi
sion 1-8-0
```

2. Ensure that the main `IstioOperator` CR has a name and revision, and does not install a gateway:

```
# filename: control-plane-1-8-0.yaml
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  name: control-plane-1-8-0 # REQUIRED
spec:
  profile: minimal
  revision: 1-8-0 # REQUIRED
```

3. Create a separate `IstioOperator` CR for the gateway(s), ensuring that it has a name and has the `empty` profile:

```
# filename: gateways.yaml
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  name: gateways # REQUIRED
spec:
  profile: empty # REQUIRED
  revision: 1-8-0 # REQUIRED
  components:
    ingressGateways:
      - name: istio-ingressgateway
        enabled: true
```

4. Apply the files to the cluster with the following commands:

```
$ kubectl create namespace istio-system
$ kubectl apply -n istio-system -f control-plan
e-1-8-0.yaml
$ kubectl apply -n istio-system -f gateways.yam
l
```

Verify that the operator and Istio control plane are installed and running.

Upgrade with operator

Let’s assume that the target version is 1.8.1.

1. Download the Istio 1.8.1 release and use the `istioctl` from that release to install the Istio 1.8.1 operator:

   ```
   $ istio-1.8.1/bin/istioctl operator init --revi
   sion 1-8-1
   ```

2. Copy the control plane CR from the install step above as `control-plane-1-8-1.yaml`. Change all instances of `1-8-0` to `1-8-1` in the files.

3. Apply the new file to the cluster:

   ```
   $ kubectl apply -n istio-system -f control-plan
   e-1-8-1.yaml
   ```

4. Verify that two versions of `istiod` are

running in the cluster. It may take several minutes for the operator to install the new control plane and for it to be in a running state.

```
$ kubectl -n istio-system get pod -l app=istiod
NAME                                READY   STATUS
   RESTARTS   AGE
istiod-1-8-0-74f95c59c-4p6mc    1/1     Running
   0          68m
istiod-1-8-1-65b64fc749-5zq8w   1/1     Running
   0          13m
```

5. Refer to the canary upgrade docs for more details on rolling over workloads to the new Istio version:
    - Label workload namespaces with istio.io/rev=1-8-1 and restart the workloads.
    - Verify that the workloads are injected with the new proxy version and the cluster is functional.
6. Upgrade the gateway to the new

revision. Edit the `gateways.yaml` file from the installation step to change the revision from `1-8-0` to `1-8-1` and re-apply the file:

```
$ kubectl apply -n istio-system -f gateways.yaml
```

7. Verify that the gateway is running at version 1.8.1.

```
$ kubectl -n istio-system get pod -l app=istio-ingressgateway --show-labels
NAME                                    READY   STATUS    RESTARTS   AGE   LABELS
istio-ingressgateway-66dc957bd8-r2ptn   1/1     Running   0          14m   app=istio-ingressgateway,service.istio.io/canonical-revision=1-8-1
...
```

8. Uninstall the old control plane:

```
$ kubectl delete istiooperator -n istio-system control-plane-1-8-0
```

9. Verify that only one version of `istiod` is

running in the cluster.

```
$ kubectl -n istio-system get pod -l app=istiod
NAME                                READY   STATUS
   RESTARTS   AGE
istiod-1-8-1-65b64fc749-5zq8w   1/1     Running
   0          16m
```